Online Normalization Algorithm for Engine Turbofan Monitoring

Jérôme Lacaille[1], Anastasios Bellas[2]

[1]*Snecma, 77550 Moissy-Cramayel, France*
jerome.lacaille@snecma.fr

[2]*SAMM, Université Panthéon-Sorbonne, 75013 Paris, France*
anastasios.bellas@malix.univ-paris1.fr

ABSTRACT

To understand the behavior of a turbofan engine, one first needs to deal with the variety of data acquisition contexts. Each time a set of measurements is acquired, and such set may account for tens of parameters, the aircraft evolves in a specific flight mode. A diagnostic of the engine behavior models the observations and tests if anything appears as expected. A model of the engine measurement vector may be very complex to produce and even more to deploy on board. The idea is to solve the problem locally on recurrent phases on which each single problem may be easier to answer. Civil flight missions are straightforward to decompose as they are very recurrent. It is more difficult with military missions and bench tests. Once a set of phases is defined, local regression models may be built. To solve nonlinearities a selection of computed variables is a good approach but such algorithm needs the definition of a stable set of recurrent phases and a very complex learning procedure that uses a huge amount of memory to deal with the high dimensionality of the problem. Such algorithm is very powerful but is not adapted for an online use. Our new solution does not require the a priori knowledge of recurrent phases; it learns recurrent contexts on the fly and adapts a small local regression model on a selected optimal subspace. The application of this algorithm seems to be efficient on long term flight trend monitoring and on real time test bench measurements. It solves the memory problem for calibration by an iterative autoadaptive procedure and suppress the need of preliminary computations of specific parameter as it auto-adapts itself with piecewise linear models.

1. INTRODUCTION

Turbofan engine abnormality diagnosis uses three steps: reduction of dependencies from the flight context (1), representation of the measurement in an adequate metric space suitable for classification and statistic testing (2) and finally identification of abnormal behavior (3) as represented on Figure 1 (next page). This work essentially deals with the first normalization step.

The current text focus on identification of flight phases to extract subsamples of temporal observations where the turbofan gross behavior may be explained by simple (eventually linear) models. This example is easy to visualize, but we also use the same algorithm on different applications. At component level we monitor the start system (Flandrois, Lacaille, Massé, & Ausloos, 2009; Lacaille, 2009), the fuel system and other turbofan components. Even to monitor bench test cells we look at vibration monitoring according to load parameters and lots of different other configurations (Lacaille & Gerez, 2011, 2012).

The first step of the algorithm is to get rid of acquisition context. This is mandatory because we need to compare similar events, observations corresponding to one unique and standard context. For this purpose we use a normalization algorithm (Figure 1, step 1). The classical method is to use a model of the engine observation measurements named endogenous parameters according to the flight context also referred as exogenous parameters (see Table 1 for a list of parameter examples). The residual between real endogenous parameters and the model results is then used as inputs to a scoring algorithm (Figure 1, step 2) which is essentially a statistical test that measures the likelihood of the current observation. The main problem is the construction of such residual. As the engine behavior is definitely nonlinear according to the flight measurements a suggestion is to cut the flight in recurrent phases: taxi, takeoff, climb, cruise, descent, etc. and models the behavior locally on those phases. However as such decomposition seems easy to build on civil mission it is a real challenge on

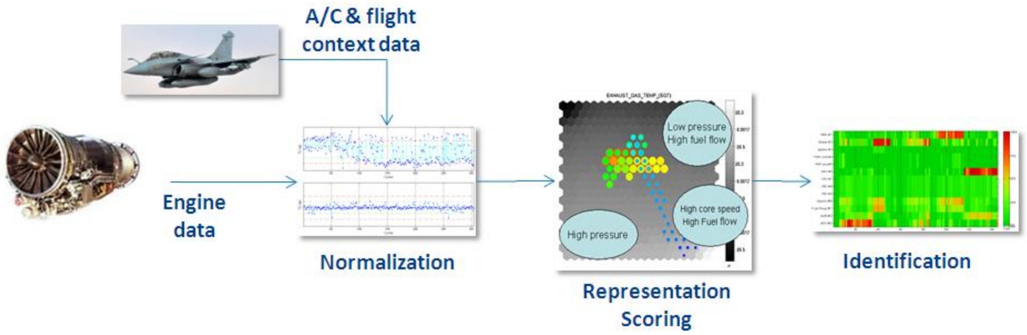

Figure 1 – The mains steps of any diagnostic application for aircraft engine monitoring.

military missions which all are different as well as for helicopter missions, business jets and event test bench tests.

Table 1 – Example of context information and endogenous measurements. Context parameters are mainly commands that describe the current engine use but also aircraft attitude. Endogenous measurements represent the observation we are really interested in to describe the engine behavior if the context was always the same during acquisition

Name	Description
Index information	
AC_ID	*Aircraft ID*
ESN	*Engine Serial Number*
FL_DATE	*Flight Date*
Context information	
TAT	*External temperature*
ALT	*Altitude*
AIE	*Anti Ice Engine*
AIW	*Anti Ice Wings*
BLD	*Bleed valve position*
ISOV	*ECS Isolation Valve Position*
VBV	*Variable Bleed Valve Position*
VSV	*Variable Stator Vane Position*
HPTACC	*High Pressure Turbine Active Clearance Control*
LPTACC	*Low Pressure Turbine Active Clearance Control*
RACC	*Rotor Active Clearance Control*
ECS	*Environmental Control System*
TLA	*Thrust Lever Angle*
N1	*Fan Speed*
XM	*Mach Number*
Endogenous measurements	
N2	*Core Speed*
FF	*Fuel Flow*
PS3	*Static pressure after compression*
T3	*Temperature after compression*
EGT	*Exhaust Gas Temperature*

Our first approaches uses manual extraction of flight phases for civil engines and a LASSO algorithm for the selection of pertinent analytical combinations of parameters to build the regression model and then a autoadaptive clustering method that uses a self-organizing map (SOM) to identify the different faults or behavior differences (Figure 1, step three “identification”). This work was presented in previous work (Côme, Cottrell, Verleysen, & Lacaille, 2010, 2011; Cottrell et al., 2009; Lacaille & Côme, 2011).

Even when flight mode identification of recurrent phases is clear, the normalization model that currently uses a LASSO regression algorithm needs a very huge amount of memory. The LASSO algorithm needs a matrix of the parameter measurements in memory: as an example the data for one engine from a set of 500 medium range flights with 100 parameters weight around 1.5 Gb when acquired at 1 Hz. Even this volume of data is not easily manageable with classical tools and standard algebraic operations such as singular values decomposition (SVD) which is the base tool in linear compression. Hence it is only possible to calibrate this model on ground on a subsample of data we may download from a small subset of aircrafts which owners (the airlines or military) let us have access to their digital flight data recorders (DFDR, the black boxes). The resulting model transferred on each engine is finally a general approximation. It misses the specificity of each engine or event the particular way each company and pilot operates its aircrafts.

2. STATISTIC MIXTURE MODEL

To solve our normalization problem iteratively with not too much memory resources involved we used a mixture of probabilistic principal component analysis (MPPCA) model. Such model is an extension of the classical PCA which goal is to extract a reduced number of dimensions on which the data may be explained. The reduction of dimension enables the computation of meaningful distances[1] and allows the

[1] Distances are needed to compute a score based on the likelihood of the difference between observation and model estimation. In high dimension,

computation of scores. However if the general behavior of observations is not linear a classical PCA algorithm will fail. A nice solution is to make the hypothesis that in each flight mode, the local behavior of the engine may have a linear representation. The MPPCA algorithm will use EM (expectation/maximization) optimization scheme to identify the clusters and to build the local projections.

We consider that we dispose of a datastream $D = \{\mathbf{x}_1, \dots \mathbf{x}_N\} \in R^p$, where the $\mathbf{x}$ are independent realizations of a random vector $X \in R^p$. In addition, $\{z_1, \dots, z_N\}$ are assumed to be independent realizations of an unobserved (latent) random variable Z with values in $\{1, \dots, K\}$ (there exists K different modes.) The MPPCA model assumes that the observed random vector $X \in R^p$ is, conditionally to Z, linked to a d-dimensional latent random vector $Y \in R^d$ through a linear transformation of the form:

$$X_{|Z=k} = U_k Y + \boldsymbol{\mu}_k + \epsilon \tag{1}$$

where U_k is the $p \times d$ orthogonal transformation matrix, $\mu_k \in R^p$ is the mean vector of the k-th factor analyzer and $\epsilon \in R^p$ is a noise term. The dimension d of the latent vector is such that $d < p$ and assumed to be known (Figure 2 below shows an illustration of K=2 d=2D subspaces in a p=3D domain).

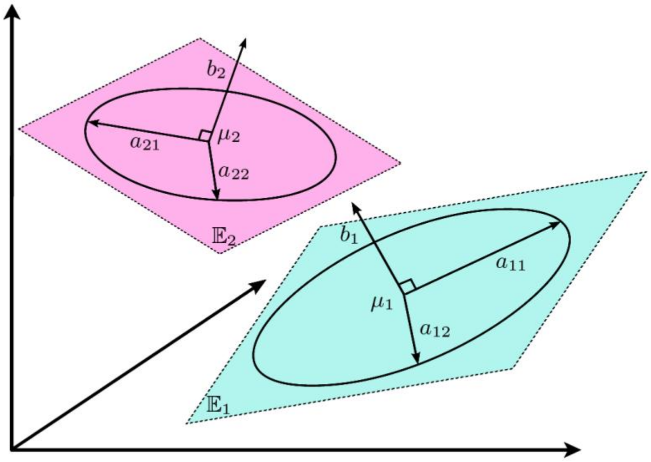

Figure 2 – Illustration of two Gaussian d=2 subspaces in a main p=3 dimension space.

Moreover, ϵ is assumed to be, conditionally to Z, a centered Gaussian noise term with a diagonal covariance matrix $b_k I_p$:

$$\epsilon_{|Z=k} = \mathcal{N}\big(0, b_k I_p\big). \tag{2}$$

Besides, the unobserved latent factor $Y \in R^d$ is assumed to be, conditionally to Z, distributed according to a Gaussian density function such as:

$$Y_{|Z=k} = \mathcal{N}(0, I_d). \tag{3}$$

This implies that the conditional distribution of X is also Gaussian:

$$X_{|Y,Z=k} \sim \mathcal{N}(U_k Y + \boldsymbol{\mu}_k, b_k I_p) \tag{4}$$

and its marginal distribution is therefore a mixture of Gaussians:

$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \phi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k) \tag{5}$$

where π_k is the mixture proportion for the k-th component, ϕ is the multivariate Gaussian density function

$$\begin{aligned} \phi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{p}{2}} \det(\Sigma_k)^{\frac{1}{2}}} \times \\ \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)' \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right) \end{aligned} \tag{6}$$

and $\Sigma_k = U_k' U_k + b_k I_p$.

In order to facilitate the description of our online inference procedure, let us slightly re-parameterize the above model. Let us first introduce the orthonormal transformation matrix Q_k which is such that its j-th column $q_{kj} = u_{kj} / \|u_{kj}\|$ where u_{kj} is the corresponding column of U_k . If the transformation matrix Q_k is orthonormal, it is then necessary to report the variance of the latent factor within the distribution of the latent factor.

We therefore now assume that $Y_{|Z=k} = \mathcal{N}(0, \Delta_k)$ where $\Delta_k = \mathrm{diag}(\lambda_{k1}, \dots \lambda_{kd})$. The marginal distribution of X is then still a mixture of Gaussians but with covariance matrices $\Sigma_k = Q_k' \Delta_k Q_k + b_k I_p$. By denoting by $Q_k = [Q_k, R_k]$ the $p \times p$ matrix made of Q_k and an orthonormal complementary R_k, the projected covariance matrix $Q_k \Sigma_k Q_k'$ has the following form:

$$\left(\begin{array}{cc} \begin{bmatrix} a_{k1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{kd} \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} b_k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_k \end{bmatrix} \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{bmatrix} a \\ \vdots \\ 0 \end{bmatrix}}\right\} \quad d \\ \left.\vphantom{\begin{bmatrix} b \\ \vdots \\ 0 \end{bmatrix}}\right\} \quad (p-d) \end{array}$$

where $a_{kj} = \lambda_{kj} + b_k$ and $a_{kj} > b_k$, for $k = 1, \dots K$ and $j = 1, \dots d$. With these notations, the mixture of PCA model is fully parameterized by the set of parameters $= (\theta_k)_{k=1\dots K}$ with each $\theta_k = \{\pi_k, \mu_k, Q_k, (a_{kj})_{j=1\dots d}, b_k\}$.

It can be shown (Bouveyron, Girard, & Schmid, 2007) that the MPPCA model is identifiable and its inference can be done using a simple EM algorithm. In particular, the update

distances lose their signification which is also known as the curse of high dimensions. We try to limit ourselves to a selected dimension smaller than 5.

formula in the M step for the orientation matrices Q_k and the variance parameters a_{kj} and b_k are as follows:

- the d columns of Q_k are estimated by the eigenvectors associated with the d largest eigenvalues of the empirical covariance matrix S_k of the k-th group,
- the empirical covariance matrix of the k-th group is $S_k = \frac{1}{n}\sum_{i=1\dots n} t_{ik}(x_i - \mu_k)(x_i - \mu_k)'$ where at each current step $t_{ik} = P(Z = k|\mathbf{x}_i;\theta)$.
- a_{kj} is estimated by the j-th largest eigenvalues of S_k,
- b_k is estimated by the residual variance $b_k = \frac{1}{p-d}\left(\mathrm{tr}(S_k) - \sum_{j=1}^{d} a_{kj}\right)$.

In addition, these update formulas illustrate the strong link between MPPCA and the principal component analysis (PCA) method, since they both consider eigenvectors corresponding to the largest eigenvalues of the covariance matrix eigen decomposition.

3. ONLINE INFERENCE OF PARAMETERS

A standard way to estimate model parameters in parametric mixture models is to maximize the (observed) log-likelihood of the data.

$$L(\mathbf{x};\theta) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \phi(\mathbf{x}_i, \theta_k) \tag{7}$$

Note that we prefer the log-likelihood over the likelihood, as it is much more convenient to work with the former from a mathematical point of view. The maximum likelihood method then proposes to estimate the parameters of the model θ by $\hat{\theta}_{MV} = \arg\max L(x;\theta)$.

As we saw earlier, complete data $\{(\mathbf{x}_1, z_1), (\mathbf{x}_2, z_2), \dots, (\mathbf{x}_n, z_n)\}$ are composed of pairs of data $\mathbf{x}$ and class information z. The complete log-likelihood $L_c(\mathbf{x}, z;\ \theta)$ is the log-likelihood calculated from the complete data:

$$L_c(\mathbf{x}, z;\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} t_k^{(i)} \log\left(\pi_k \phi(\mathbf{x}_i, \theta_k)\right) \tag{8}$$

Here, we have defined t as the indicator variable of the classes, so that if $z_i = k$ for a data sample i, then $t_k^{(i)} = 1$ and $t_k^{(j)} = 0, \forall j \neq k$.

In order to extend MPPCA to the online setting, we develop hereafter an online EM-based algorithm which incorporates a probabilistic version of the incremental PCA (Hall, Marshall, & Martin, 1998). We consider here a setting where data samples are arriving in an online manner and each data sample is being discarded after being processed (Bellas, Bouveyron, Cottrell, & Lacaille, 2013).

Let us assume that we initially have observed a dataset of n_0 data samples $(\mathbf{x}_1 \dots \mathbf{x}_{n_0}) \in R^p$ and that we have obtained an initial estimate $\hat{\theta}^{(n_0)}$ of these data. In practice, we obtain an initial estimation of the model parameters with a standard MPPCA iterative EM algorithm on this initial dataset. Let us set $n = n_0$ and consider the arrival of a new data sample $\mathbf{x}_{n+1} \in R^p$.

The objective is therefore to update the estimate of θ from the sole knowledge of $\hat{\theta}^{(n)}$ and $\mathbf{x}_{n+1}$. This is a two-step procedure which involves an expectation step (E-step) and a maximization step (M-step).

3.1. The E-step

Before updating the estimate of θ, it is necessary to compute the expectation of the complete log-likelihood $E\left(L_c(\mathbf{x}, z;\theta)\middle|\theta^{(n)}\right)$ conditionally to the current estimate $\hat{\theta}^{(n)}$.

This quantity will be maximized in the second step to obtain the new estimate $\hat{\theta}^{(n+1)}$ of θ. As with all mixture models, the computation of the conditional expectation of the complete log-likelihood reduces, in the context of the MPPCA model, to the computation of the probabilities $t_k^{(n+1)} = P\left(Z = k\middle|X = \mathbf{x}_{(n+1)}\right)$ that the new data sample belongs to the k-th mixture component (Figure 3). These probabilities can be computed as follows:

$$\begin{aligned} t_k^{(n+1)} &= \frac{\pi_k \phi\left(x_{n+1};\ \hat{\theta}_k{}^{(n)}\right)}{\sum_{l=1}^{K} \pi_l \phi\left(x_{n+1};\ \hat{\theta}_l{}^{(n)}\right)} \\ &= 1\Big/\sum_{l=1}^{K} \exp\left(\tfrac{1}{2}\left(\Gamma_k^{(n)}(x_{n+1}) - \Gamma_l^{(n)}(x_{n+1})\right)\right) \end{aligned} \tag{9}$$

where the classification function Γ_k has the following form:

$$\begin{aligned} \Gamma_k(x) &= \|\boldsymbol{\mu}_k - P_k(\mathbf{x})\|_{\mathcal{A}_k}^2 + \tfrac{1}{b_k}\|\mathbf{x} - P_k(\mathbf{x})\|^2 \\ &\quad + \sum_{j=1}^{d} \log\left(a_{kj}\right) + (p-d)\log(b_k) - 2\log(\pi_k) \end{aligned} \tag{10}$$

$$\text{with} \begin{cases} \|\mathbf{x}\|_{\mathrm{A_k}}^2 &= \mathbf{x}'\mathrm{A_k}\mathbf{x} \\ \mathrm{A_k} &= \mathrm{Q_k}\Delta_{\mathrm{k}}^{-1}\mathrm{Q}_{\mathrm{k}}' \\ \mathrm{P_k}(\mathbf{x}) &= \mathrm{Q_k}\mathrm{Q}_{\mathrm{k}}'(\mathbf{x}\text{-}\boldsymbol{\mu}_{\mathrm{k}})+\boldsymbol{\mu}_{\mathrm{k}}. \end{cases}$$

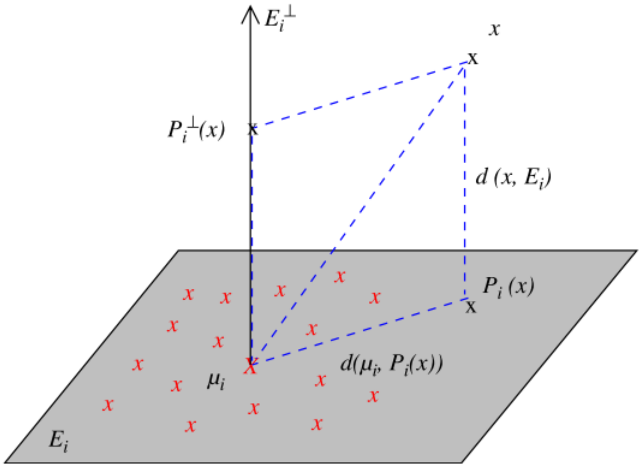

Figure 3 – Geometric interpretation of the probability that a sample belongs to a given class.

3.2. The M-step

Once the posterior probabilities $t_k^{(n+1)}$ have been computed, we update the model parameters so that they maximize $E\big(L_c(\mathbf{x}, z; \theta)\big|\theta^{(n)}\big)$. In order to derive an online inference strategy which does not keep all past data samples, it is necessary to make use of the following approximation:

$$E\big(L_c(\mathbf{x}, z; \theta)\big|\theta^{(n)}\big) \simeq E\big(L_c(\mathbf{x}, z; \theta)\big|\theta^{(n-1)}\big) + \sum_{k=1}^{K} t_k^{(n+1)} \log\Big(\pi_k \phi\big(\mathbf{x}_{n+1}; \theta_k^{(n)}\big)\Big) \tag{11}$$

Then, it is straightforward to show that the update formulas for the mixture proportions π_k and the component means $\boldsymbol{\mu}_k$, for every component $k = 1 \ldots K$, are:

$$\begin{aligned} \pi_k^{(n+1)} &= \pi_k^{(n)} + \tfrac{1}{n+1}\big(t_k^{(n+1)} - \pi_k^{(n)}\big), \\ \boldsymbol{\mu}_k^{(n+1)} &= \tfrac{1}{n_k^{(n+1)}}\big(n_k^{(n)}\boldsymbol{\mu}_k^{(n)} - t_k^{(n+1)}\mathbf{x}_{n+1}\big), \end{aligned} \tag{12}$$

where $n_k^{(n+1)} = n_k^{(n)} + t_k^{(n+1)}$ and $n = \sum_{k=1\ldots K} n_k^{(n)}$.

We then want to estimate the parameters Q_k, a_{kj} and b_k, for $k = 1 \ldots K$ and $j = 1 \ldots d$. We have already seen that the maximization of $E\big(L_c(\mathbf{x}, z; \theta)\big|\theta^{(n)}\big)$ with respect to these parameters is equivalent to the eigen decomposition of the empirical covariance matrix S_k for each component $k = 1 \ldots K$. The problem that we seek to solve can be therefore stated as follows: having already calculated eigenvectors $Q_k^{(n)}$ and eigenvalues $\Lambda_k^{(n)}$ from the n first data samples, we want to update those parameters on the arrival of a (n+1)-th data sample. In particular, on the arrival of the new data sample $\mathbf{x}_{n+1}$, the new eigenproblem that we need to solve is:

$$\Sigma_k^{(n+1)} Q_k^{(n+1)} = Q_k^{(n+1)} \Lambda_k^{(n+1)} \tag{13}$$

where $\Lambda_k^{(n+1)} = diag\{\lambda_{k1} \ldots \lambda_{kp}\}$ and this for $k = 1 \ldots K$. To begin with, let us define:

$$\begin{aligned} g_k^{(n+1)} &= \big(Q_k^{(n)}\big)'\big(t_k^{(n+1)}\mathbf{x}_{n+1} - \boldsymbol{\mu}_k^{(n)}\big) \\ h_k^{(n+1)} &= \big(t_k^{(n+1)}\mathbf{x}_{n+1} - \boldsymbol{\mu}_k^{(n)}\big) - Q_k^{(n)} g_k^{(n+1)} \end{aligned} \tag{14}$$

where $g_k^{(n+1)}$ is the projection of the data sample on the subspace defined by the eigenvectors and $h_k^{(n+1)}$ is the residue of the retro-projection on the original space. With these notations, the new eigenvectors $Q_k^{(n+1)}$ correspond to a rotation of the old ones plus the unit residue vector $\tilde{h}_k^{(n+1)}$:

$$\tilde{h}_k^{(n+1)} = \begin{cases} \dfrac{h_k^{(n+1)}}{\big\|h_k^{(n+1)}\big\|_2}, \text{ if } \big\|h_k^{(n+1)}\big\|_2 \neq 0 \\ 0, \text{otherwise.} \end{cases} \tag{15}$$

and thus the new eigenvectors may be written:

$$Q_k^{(n+1)} = \big[Q_k^{(n)}, \tilde{h}_k^{(n+1)}\big] R_k^{(n+1)} \tag{16}$$

where $R_k^{(n+1)}$ is a rotation matrix of size (d+1)×(d+1). Note that $Q_k^{(n)}$ is a $p \times d$ matrix, since we have discarded the $p-d$ less significant eigenvalues. The new covariance matrix $\Sigma_k^{(n+1)}$ for the class k is given by:

$$\Sigma_k^{(n+1)} = \frac{n_k^{(n)}}{n_k^{(n+1)}} \Sigma_k^{(n)} + \frac{n_k^{(n)}}{\big(n_k^{(n+1)}\big)^2} \bar{\mathbf{x}}\bar{\mathbf{x}}' \tag{17}$$

where we have set $\bar{\mathbf{x}} = t_k^{(n+1)}\mathbf{x}_{n+1} - \boldsymbol{\mu}_k^{(n+1)}$. Then, by substituting equations (16) and (17) into equation (13) we get[2]:

$$\big[Q_k^{(n)}, \tilde{h}_k\big]' \left(\frac{n_k^{(n)}}{n_k^{(n+1)}} \Sigma_k^{(n)} + \frac{n_k^{(n)}}{\big(n_k^{(n+1)}\big)^2} \bar{\mathbf{x}}\bar{\mathbf{x}}' \right) \big[Q_k^{(n)}, \tilde{h}_k\big] R_k^{(n+1)} = R_k^{(n+1)} \Lambda_k^{(n+1)} \tag{18}$$

The above problem can be written as:

$$\left(\frac{n_k^{(n)}}{n_k^{(n+1)}} \begin{bmatrix} \Lambda_k^{(n)} & 0 \\ 0 & 0 \end{bmatrix} + \frac{n_k^{(n)}}{\big(n_k^{(n+1)}\big)^2} \begin{bmatrix} g_k g_k{}' & \gamma_k g_k \\ \gamma_k g_k{}' & \gamma_k^2 \end{bmatrix} \right) R_k^{(n+1)} = R_k^{(n+1)} \Lambda_k^{(n+1)} \tag{19}$$

where we have set $\gamma_k^{(n+1)} = \tilde{h}_k^{(n+1)\prime}\bar{\boldsymbol{x}}$. The solution to this new eigenproblem yields the rotation matrix $R_k^{(n+1)}$ and the new eigenvalues $\Lambda_k^{(n+1)}$ directly. Then, the new eigenvectors can be obtained using equation (16). Note that

[2] For simplicity we omit temporal subscript $(n+1)$ for vectors h_k and g_k.

both $R_k^{(n+1)}$ and $\Lambda_k^{(n+1)}$ are square matrices of dimension d+1, that is, we only need to solve an eigenproblem of dimension d+1 and not p. The update formulas for the variance parameters a_{kj} and b_k are then:

$$a_{kj}^{(n+1)} = \Lambda_{kj}^{(n+1)},$$
$$b_k^{(n+1)} = \frac{1}{p-d}\left(\mathrm{tr}\left(\Lambda_k^{(n+1)}\right) - \sum_{j=1}^{d} a_{kj}^{(n+1)}\right). \tag{20}$$

4. COMPARISONS WITH ONLINE EM AND CEM

We compare online MPPCA with two other online algorithms, online EM (Titterington, 1984) and online CEM (Samé, Ambroise, & Govaert, 2007). Note that these latter have not been designed to handle high-dimensional data. This benchmark was done on simulated data because we could control the real problem dimension which is not the case with real observations. An application on turbofan engine measurements is given in the next section.

For this experiment, we have generated a dataset of n = 12000 data samples in R^p based on the assumption that data live in low-dimensional subspaces, with p = 30 and K = 3. Hereafter, we refer to this dataset as D_{30}. The mixture proportions are $\pi_1 = 0.4$ and $\pi_2 = \pi_3 = 0.3$. For simplicity, we have considered that for each class, the variance is common across all dimensions, that is $a_{kj} = a_k$, for $k = 1 \dots K$ and $j = 1 \dots d$. We have set $a_1 = 150$, $a_2 = 75$, $a_3 = 50$, $b_1 = b_2 = b_3 = 5$ and $\mu_1 = 0$, $\mu_2 = \{0\dots5\dots0\}$ and $\mu_3 = \{0\dots{-5}\dots0\}$, with $\mu_1, \mu_2, \mu_3 \in R^p$. We have set the intrinsic dimension (dimension of the subspaces) at $d = 2$.

We also simulate a second dataset of lower dimension (p = 10) , generated with the same parameters as the former. We will refer to this new dataset as D_{10}.

Our goal was to study the behavior of the three algorithms in low dimension and then illustrate the capability of online MPPCA to cluster efficiently even in high dimension.

We have evaluated the three algorithms on the quality of their estimation of the class means and on the accuracy of the clustering produced. The quality of the estimation of the means was taken to be the square of the distance of the estimated means to the true ones, averaged over all K = 3 classes, a measure known as the Mean Square Error (MSE) in statistics

$$\mathrm{MSE}_\mu = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{p}\sum_{j=1}^{p}\left(\hat{\boldsymbol{\mu}}_{kj} - \boldsymbol{\mu}_{kj}\right)^2\right) \tag{21}$$

Online MPPCA, online EM and online CEM were initialized 30 times by a standard MPPCA, an EM and a CEM, respectively, of which the initialization giving the highest BIC value was kept.

Figure 4 and Figure 5 show the comparative performance of online MPPCA (black), online EM (red) and online CEM (blue) for the datasets D_{10} and D_{30}, respectively.

For the dataset D_{10} it is clear, both from the clustering accuracy and the MSE that online MPPCA converges faster than the other two algorithms.

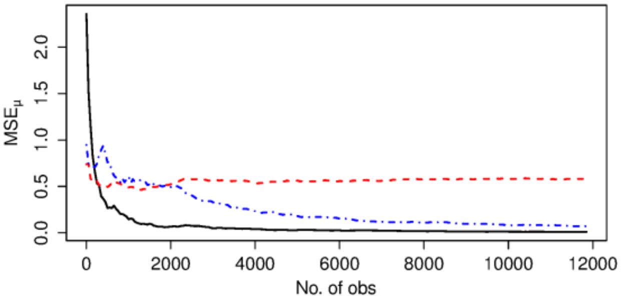

Figure 4 – Evolution of MSE for the dataset D_{10} versus the number of data samples for online MPPCA (black solid), online EM (red dashed) and online CEM (blue dotted).

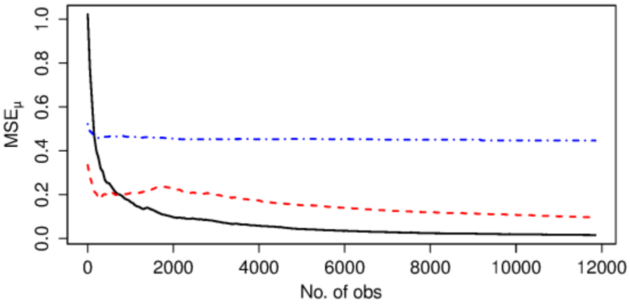

Figure 5 – Evolution of MSE for the dataset D_{30} versus the number of data samples for online MPPCA (black solid), online EM (red dashed) and online CEM (blue dotted).

5. APPLICATION TO ENGINE HEALTH MONITORING

We test the proposed method to real data issued from the aircraft engine Health Monitoring domain. The data were obtained by Snecma.

Typically, there exists different phases during a flight, called flight modes: taking-off, cruising, landing etc. Each test is actually a sequence of alternating stationary and non-stationary phases at different levels. The stationary phases correspond in general to such flight modes, while the non-stationary ones reflect the transition between two such phases. Nevertheless, a flight mode can include multiple stationary phases, that is, a stationary control on the data is not enough to detect the flight modes.

Aircraft engineers can identify these modes by looking at the data but this can be extremely time-consuming. Moreover, due to the high dimensionality of data, there can be relations that humans cannot perceive. Note that by knowing, at any given time, in which flight mode the engine currently is, tasks like anomaly detection can be performed much more reliably, since the 'local' context of the data is also taken into account.

The experiment below (Figure 6) involves a MPPCA stage used to build a residual vector that is finally classified with a self organizing map (SOM). The score represents the distance to the corresponding class center, and the fault identification is obtained as the map cell number.

The data simulates real engine normal degradation (usual wear) to be detected by trend monitoring tools. The result appears to be pertinent for operational analysis as the MRO operator usually waits for confirmation before any customer notification.

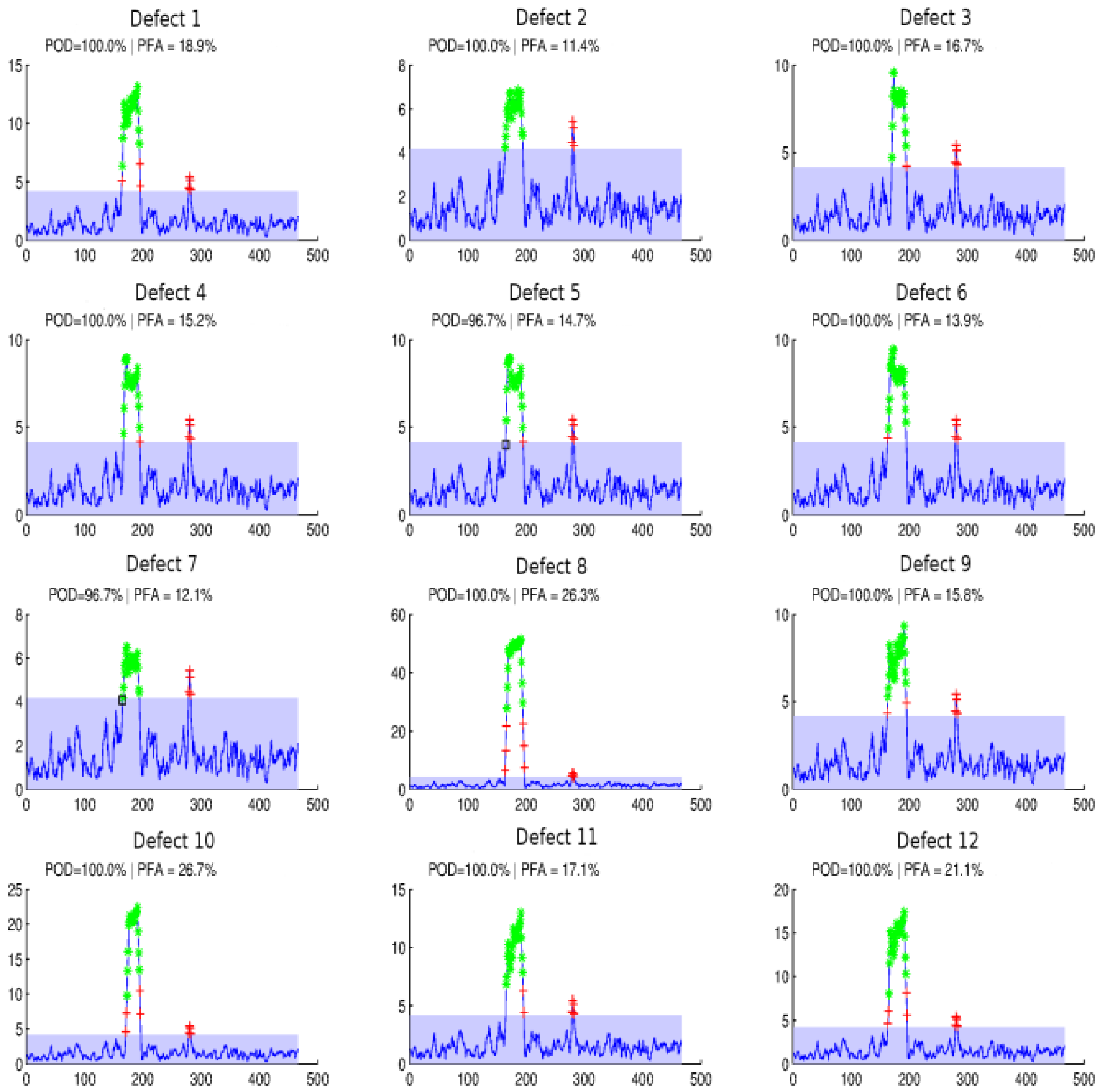

Figure 6 – Scoring and identification of trend faults using a self organizing map after MPPCA normalization. Green dots are the true detections and red ones the false alarms. POD stands for probability of detection and is given as a point to point count, as well as the PFA which is the probability of false alarm.

6. CONCLUSION

We have proposed an online inference algorithm for the MPPCA model which relies on an EM-based procedure and a probabilistic and incremental version of PCA. The proposed strategy allows to incrementally update the estimates of the MPPCA parameters at the arrival of a new data sample. It allows also providing low-dimensional visualizations of the data based on sufficient information. Model selection is also considered in the online setting through parallel computing. Numerical experiments on simulated and real data have shown that the online MPPCA algorithm performs better in high-dimensional spaces compared to existing online EM-based algorithms.

NOMENCLATURE

ACARS	Aircraft Communications Addressing and Reporting System
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
DFDR	Digital Flight Data Recorder
EM	Expectation Maximization
LASSO	Least Absolute Shrinkage and Selection Operator
MPPCA	Mixture of Probabilistic PCA
MRO	Maintenance Repair Overhaul
MSE	Mean Square Error
PCA	Principal Component Analysis
SOM	Self Organizing Map

REFERENCES

Bellas, A., Bouveyron, C., Cottrell, M., & Lacaille, J. (2013). Model-based Clustering of High-dimensional Data Streams with Online Mixture of Probabilistic PCA. *ADAC*.

Bouveyron, C., Girard, S., & Schmid, C. (2007). High-dimensional data clustering. *Computational Statistics & Data Analysis*, *52*(1), 502–519. doi:10.1016/j.csda.2007.02.009

Côme, E., Cottrell, M., Verleysen, M., & Lacaille, J. (2010). Aircraft engine health monitoring using Self-Organizing Maps. In *Industrial Conference on Data Mining*. Berlin.

Côme, E., Cottrell, M., Verleysen, M., & Lacaille, J. (2011). Aircraft engine fleet monitoring using Self-Organizing Maps and Edit Distance. In *WSOM*. Espoo (Finland).

Cottrell, M., Gaubert, P., Eloy, C., François, D., Hallaux, G., Lacaille, J., & Verleysen, M. (2009). Fault prediction in aircraft engines using Self- Organizing Maps. In *WSOM*. Miami (FL).

Flandrois, X., Lacaille, J., Massé, J.-R., & Ausloos, A. (2009). Expertise Transfer and Automatic Failure Classification for the Engine Start Capability System. In *AIAA InfoTech*.

Hall, P., Marshall, A., & Martin, R. (1998). Incremental Eigenanalysis for Classification. *BMVC*, 29.1–29.10. doi:10.5244/C.12.29

Lacaille, J. (2009). Standardized failure signature for a turbofan engine. In *IEEE Aerospace conference* (p. 11/0505). Big Sky (MT): IEEE Aerospace society. doi:10.1109/AERO.2009.4839670

Lacaille, J., & Côme, E. (2011). Visual Mining and Statistics for a Turbofan Engine Fleet. In *IEEE Aerospace Conference* (p. 11/0405). Big Sky (MT): IEEE.

Lacaille, J., & Gerez, V. (2011). Online Abnormality Diagnosis for real-time Implementation on Turbofan Engines and Test Cells. In *PHM*. Montreal (Canada): PHMSociety.

Lacaille, J., & Gerez, V. (2012). A Batch Detection Algorithm Installed on a Test Bench. In *PHM* (pp. 1–7). Minneapolis: PHM Society.

Samé, A., Ambroise, C., & Govaert, G. (2007). An online classification EM algorithm based on the mixture model. *Statistics and Computing*, *17*(3), 209–218. doi:10.1007/s11222-007-9017-z

Titterington, D. M. (1984). Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society. Series B (Methodological)*, *46*(2), 257–267.

BIOGRAPHY

Jérôme Lacaille is a Safran emeritus expert which mission for Snecma is to help in the development of mathematic algorithms used for the engine health monitoring. Jérôme has a PhD in Mathematics on “Neural Computation” and a HDR (habilitation à diriger des recherches) for “Algorithms Industrialization” from the Ecole Normale Supérieure (France). Jérôme has held several positions including scientific consultant and professor. He has also co-founded the Miriad Technologies Company, entered the semiconductor business taking in charge the direction of the Innovation Department for Si Automation (Montpellier - France) and PDF Solutions (San Jose - CA). He developed specific mathematic algorithms that where integrated in industrial process. Over the course of his work, Jérôme has published several papers on integrating data analysis into industry infrastructure, including neural methodologies and stochastic modeling.

Anastasios Bellas is a mathematic PhD form university Paris 1 Pantheon-Sorbonne. He worked on online autoadaptive clustering methodologies applied to aircraft engine measurements for Snecma. Today, Anthikos is doing is military service in Greece.